Deadline Constrained Scheduling for Data Aggregation in Unreliable Sensor Networks

Srikanth Hariharan and Ness B. Shroff

Abstract—**We study the problem of maximizing the aggregated information in a wireless sensor network. We consider a sensor network with a tree topology, where the root corresponds to the sink, and the rest of the network detects an event and transmits data to the sink. We formulate an integer optimization problem that maximizes the *aggregated information* that reaches the sink under deadline and interference constraints. This framework allows using a variety of error recovery schemes to tackle link unreliability. We show that the optimal solution involves solving a Job Interval Selection Problem (JISP) which is known to be MAX SNP-Hard. We construct a sub-optimal version, and develop a low complexity, distributed optimal solution to this version. We investigate tree structures for which this solution is optimal to the original problem. Our numerical results show that the sub-optimal solution outperforms existing JISP approximation algorithms even for general trees.**

I. Introduction

A wireless sensor network is a wireless network consisting of a number of sensors that sense a desired aspect of the region in which they are deployed. These networks are used in a number of military and civilian applications, such as target tracking and environment monitoring. Sensor measurements are prone to errors due to environmental factors and resource constraints. Therefore, sinks cannot rely on the data sensed by a single sensor. In many applications, the sinks only desire a certain function of the data sensed by different sensor nodes (e.g., average temperature, maximum pressure, detect a signal, etc.). When sinks require certain classes of functions of the sensed data, performing *in-network computation* (intermediate nodes in the network aggregate data from all their predecessors, and only transmit the aggregated data) greatly reduces the communication overhead [1].

A tree structure is commonly used for data aggregation in wireless sensor networks [2], [3]. In this paper, we consider a tree topology with the sink as the root of the tree. An event is observed by a subset of nodes in the tree called the source nodes. All source nodes transmit their data about the event to the sink. Our goal is to maximize the *information* obtained by the sink. The information obtained by the sink is a representation of the quality of the data that reaches the sink. For example, it could be the sum of the inverses of the error variances of the data from various sources that reaches the sink [4]. It could also represent other relevant metrics such as the Log-Likelihood Ratio if detection is being performed by the network, distortion, etc. Since the information obtained at the sink corresponds to data that has been "aggregated" within the network (as opposed to being transmitted separately from individual sources), we call the information obtained at the sink "aggregated information".

Delay is also an important parameter in a wireless sensor network. While most works focus only on energy, minimizing the delay can help save a huge amount of energy. For instance, suppose that a sensor network is tracking a target. In order to ensure good tracking quality, the sink must obtain previous measurements in a timely manner so that the best subset of sensors for the next measurement is chosen. If the sink does not get the measurements in a timely manner, the target might have moved too far resulting in a poor measurement quality during future measurements. On the other hand, if the sink decides to ensure good tracking quality by activating a large number of sensors at all times, a large amount of energy could be wasted. Therefore, it is critical that the sink obtains sensor measurements in a delay efficient manner.

The main contributions of this work are summarized as follows:

- We develop an optimization based framework to maximize the aggregated information that is received at the sink from all the source nodes in the data gathering tree. This optimization framework *explicitly accounts for unreliable links, deadlines, and interference*. Further, it allows to tackle link unreliability using a variety of error-recovery schemes as long as errors across links are independent.
- We show that solving this optimization problem involves solving a single-server Job Interval Selection Problem (JISP) which is known to be MAX SNP-Hard (which means that unless $P = NP$, a Polynomial Time Approximate Scheme (PTAS) does not exist for the problem). Note that a PTAS is an algorithm which takes an instance of an optimization problem and any $\epsilon > 0$, and produces a polynomial time solution that is within a factor ϵ of being optimal.
- We study the structural properties of this problem, and develop a sub-optimal version. We provide a low-complexity, distributed optimal solution to this sub-optimal version. This solution is optimal to the original problem for certain tree structures.

This work was supported in part by ARO MURI Awards W911NF-07-10376 (SA08-03) and W911NF-08-1-0238, and NSF Awards 0626703-CNS, 0635202-CCF, and 0721236-CNS.

S. Hariharan is with the Department of Electrical and Computer Engineering, The Ohio State University, 2015 Neil Ave., Columbus, OH 43210, USA harihars@ece.osu.edu

N. B. Shroff is with the Department of Electrical and Computer Engineering and the Department of Computer Science and Engineering, The Ohio State University, 2015 Neil Ave., Columbus, OH 43210, USA shroff@ece.osu.edu

- We evaluate the performance of the sub-optimal algorithm for general tree structures through numerical evaluations, and show that it not only performs close to the optimal solution but also outperforms existing approximation algorithms for JISP.

The rest of this paper is organized as follows. In Section II, we overview related work. In Section III, we describe our system model and assumptions. In Section IV, we formulate our problem. In Section V, we study the structural properties of the deadline constrained problem, and prove that it is MAX SNP-Hard. In Section VI, we study a sub-optimal version of the deadline constrained problem, and provide an optimal algorithm for this problem. In Section VII, we provide simulation results comparing the performance of our sub-optimal algorithm with that of the optimal solution for general tree structures. Finally, in Section VIII, we conclude the paper.

II. Related Work

The problem of evacuating all the packets in a multi-hop wireless network in minimum time is quite related to what we are studying. Different works [5], [6] have studied this problem for error-free channels. They propose polynomial-time algorithms for tree structures for the one-hop interference model. These works do not consider in-network computation.

Other existing work in this area can be broadly characterized into two classes - constructing efficient data aggregation trees [2], [7] and approaches to study the trade-offs between energy, delay and the quality of aggregated data [8], [9], [10], [11]. Constructing an optimal data aggregation tree is NP-Hard for a number of cases such as minimizing the number of transmissions [7], maximizing the lifetime [2], etc.

Several existing works use optimization approaches to study energy-delay trade-offs and energy-quality trade-offs under data aggregation. Boulis et. al., [8] study trade-offs between energy and data accuracy in data aggregation trees. In [9], Yu et. al., study trade-offs between energy and latency in data aggregation trees, assuming a time-slotted synchronized system. In [10], Ye et. al., study the fundamental energy-delay trade-off for distributed data aggregation in wireless sensor networks. Their goal is to maximize a certain parameter called the discounted reward at each node, where the reward is due to data aggregation and the discount is due to the time for which the node waits in order to aggregate data from its predecessors. The common drawback in all of these works is that they do not take link errors and interference constraints into account while framing their optimization problem. We have previously studied the problem of maximizing aggregated information in data gathering trees under deadline and one-hop interference constraints, *for error-free links* [11]. For this problem, a distributed optimal solution was developed that involved solving a localized Maximum Weighted Matching (MWM) problem at each hop. Since the matching only involved neighboring nodes, the algorithm had a low computational complexity. However, the inclusion of link errors, as done in this work, substantially increases the difficulty of the problem.

III. System Model and Assumptions

We model the system as a graph $G(V \cup \{S\}, E)$ where V is the set of *active* nodes, S is the sink, and E is the set of links. The graph G is a tree rooted at the sink. An *active* node is one that is either a source node or has at least one source node in its sub-tree. Sensors take measurements for *events* (such as tracking a target, measuring the temperature, etc.), and send an aggregated form of the data to the sink within a deadline. A node may or may not be a source for a particular event.

We make the following assumptions: (i) "perfect" aggregation - intermediate nodes can gather data from predecessors, and aggregate them into a single packet [1], [7], (ii) time-slotted and synchronized system, (iii) integer number of time slots to transmit a packet over a link, (iv) negligible time required to perform in-network computation, (v) each source has data ready at time zero (for simplicity of exposition), (vi) each source has a single data packet for an event, (vii) the next event occurs only after the deadline for the current event has expired, $(viii)$ a primary (or one-hop) interference model where no two links that share a node can be active at the same time, and (ix) unreliable links in the network where errors are independent across links.

We refer to the actual sensor measurements as *data*. We define *information* as the quality of the data. For example, data could be temperature, location, etc., while information could be error variance, distortion, Log-Likelihood Ratio, etc. By aggregated data (or information), we mean the in-network computed data (or the information provided by the in-network computed data). Let w_i represent the information provided by a source node i. w_i, for example, could represent the inverse of the predicted variance of a Kalman filter if node i is tracking a target.

Since links are unreliable, we can employ a number of known error-recovery techniques such as retransmissions, coding, etc. We model the *link reliability* of a link i as a function $f_i(n)$, where n is the number of time slots consumed on transmissions over link i. For instance, if we use retransmissions, $f_i(n)$ could denote the probability that information is successfully transmitted over link i if a maximum of n transmissions (including retransmissions) are allowed. Note that $f_i(n)$ can be any arbitrary function (that is of practical importance), and hence can model any known error-recovery technique for that particular link. Similarly, it can account for different link capacities. For instance, if 2 time slots are required to transmit data over link i, and $n = 1$, $f_i(1)$ can be set to zero.

We consider two definitions of the *information received at node B from a particular source node A*.

Definition 1: Product of the information provided by node A and the *product (or weighted product)* of the *link reliabilities* over all the links from node A to node B. For example, if $P_{A,B}$ represents the path from A to B, then the information received at B from A is $w_A \prod_{j \in P_{A,B}} f_j(n_j)$.

Definition 2: Product of the information provided by node A and the *sum (or weighted sum)* of the *link reliabilities* over all the links from node A to node B. For example, the information received at B from A (over a path $P_{A,B}$ from A to B) is $w_A(\sum_{j\in P_{A,B}} f_j(n_j))$.

We model the *aggregated information received at node* B as a (weighted) sum of the information received at B from individual source nodes, where the latter is obtained from Definition 1 or Definition 2. Note that when B is the sink, we obtain the aggregated information received at the sink. This metric is of importance especially when sensor measurements are fused. For instance, the *sum* of the inverses of the error variances of individual sensors represents the overall error variance of the fused measurements when measurements are independent [4].

IV. Problem Formulation

We now formulate our optimization problem. We first describe our forwarding policy for data aggregation.

Forwarding Policy: For an event, each node will wait for a certain time to aggregate data from its predecessors. Until that waiting time expires, a node, even if it is a source node, will not transmit its data to its parent. After the waiting time expires, the node will no longer accept transmissions from its children. The implication of this policy is that a node will never transmit data that it receives from two or more nodes separately. It will always aggregate the data that it receives and transmit the aggregated data.

We now provide a few notations and formulate our optimization problem. Let V_L bet the set of leaf nodes, and V_S be the set of source nodes. Let n_{ij} be the number of time slots allocated to node i to transmit to its parent j. Also, β_{ij} is a known constant representing the maximum number of slots that can be allocated to node i. This could depend on a number of factors such as link unreliability, energy expenditure, etc. Let W_i be the waiting time of node i (as defined in the forwarding policy).

Note that the information received at the sink from a particular source node can be obtained from either Definition 1 or Definition 2. Since the solution methodology is identical for both definitions, WLOG the rest of the paper considers Definition 1.

Problem Π_D:

$$\begin{aligned}
\max_{\vec{n},\vec{W}} \quad & \sum_{i\in V_S} \text{Information received at } S \text{ from } i \text{ (Definition 1)} \\
\text{s.t.} \quad & n_{ij} \in \{0,1,...,\beta_{ij}\}\ \forall (i,j)\in E \\
& \text{For each } i \in V\cup\{S\}\backslash V_L\text{:} \\
& \forall C \subseteq \{(j,i) : (j,i)\in E\}, \\
& \sum_{j:(j,i)\in C} n_{ji} \le W_i - \min_{j:(j,i)\in C} W_j \qquad (1) \\
& W_i \in \{0,1,...,D-1\}\ \forall i\in V \text{ and } W_S = D
\end{aligned}$$

Problem Π_D is straightforward to interpret except for constraint (1). The relation between one-hop interference and delay is represented by (1). Under the one-hop interference model, a parent node can only receive packets from one of its children nodes during a particular slot. However, when a child transmits to its parent, the other children (of the same parent) can receive data from their children (by the definition of the one-hop interference model). Constraint (1) says that for any subset of the children nodes, the total number of transmissions made by this subset of nodes is bounded above by the difference between the waiting time of the parent and the waiting time of the child that has the least waiting time in the chosen subset.

For example, consider Figure 1 with node P receiving data from its children C_1, C_2, and C_3. This figure represents a single hop in a large data aggregation tree. During a slot in which C_1 transmits to P, C_2 and C_3 can receive data from their children. However, no two children of P can transmit to P in the same slot. Let node P have a waiting time W, and C_1, C_2 and C_3 have waiting times W_1, W_2 and W_3, respectively. Let $W_1 < W_2 < W_3 < W$. Then, the total number of transmissions that can be made from all the children nodes to the parent P is limited by the difference between W and W_1 (since the first transmission can occur only after W_1 and the last transmission can occur only before W, by the definition of waiting time). Also, the total number of transmissions that can be made from C_2 and C_3 to P is limited by $W - W_2$. So, Equation (1) says that for any subset of the children nodes, the total number of transmissions made by this subset of nodes is bounded above by the difference between the waiting time of the parent and the waiting time of the child that has the least waiting time in the chosen subset.

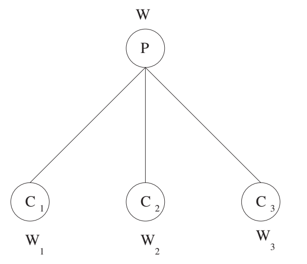


Fig. 1. Parent P waits to receive packets from children C_1, C_2 and C_3

V. Structural Properties and NP-Hardness

In this section, we study the properties of the optimal solution of problem Π_D, and show that it is NP-Hard. These properties will be used in formulating a sub-optimal problem in the next section.

Theorem V.1. *Consider any single hop in the data aggregation tree with parent node P having k children, C_1, C_2,..., C_k. For problem Π_D, let an optimal waiting time of P be W_P^* and let W_1^*,..., W_k^* be optimal waiting times of the k children, respectively. Let n_1^*, n_2^*,..., n_k^* be an optimal solution for the number of transmissions made by the k children, respectively. WLOG, assume that $W_1^* \le W_2^* \le ... \le W_k^*$. For $j \in \{1,2,...,k\}$, define W'_j recursively as follows.*

$$W'_1 = W_1^* \qquad (2)$$

$$W'_j = \max(W_j^*, W'_{j-1} + n_{j-1}^*), \text{ if } j \in \{2,3,...,k\} \qquad (3)$$

Then, for each $j \in \{1, 2, ..., k\}$, we have the following properties:

1) *W'_j is also an optimal waiting time for node C_j.*
2) *W'_j, $W'_j + 1$, ..., $W'_j + n^*_j - 1$ are optimal transmission slots for node C_j.*

Proof: We show 1) and 2) by induction on j.

$j = 1$: By definition, $W'_1 = W^*_1$, and hence W'_1 is an optimal waiting time for node C_1. We prove 2) by contradiction. Suppose that W'_1, $W'_1 + 1$, ..., $W'_1 + n^*_1 - 1$ are not optimal transmission slots for node C_1. Since C_1 cannot transmit before W'_1 because of the forwarding policy, and since C_1 needs to make n^*_1 transmissions, C_1 must transmit in at least one time slot after $W'_1 + n^*_1 - 1$. If there exists a time slot in $\{W'_1, W'_1 + 1, ..., W'_1 + n^*_1 - 1\}$ during which none of the children of P transmit, then making C_1 transmit during this slot does not affect the optimality of the solution. Suppose that there exists a time slot in $\{W'_1, W'_1 + 1, ..., W'_1 + n^*_1 - 1\}$ during which a child of P other than C_1 transmits. Once again, by making C_1 transmit during this slot, and scheduling the other child to transmit in the time slot after $W'_1 + n^*_1 - 1$ (in which C_1 was originally transmitting), the optimality of the solution is not affected. This can be reasoned as follows.

- The new schedule is feasible.
- The value of the objective function does not decrease because of interchanging the schedules. This is because the expected number of packets aggregated by C_1 does not change after time slot W'_1, and the expected number of packets aggregated by any other node in that hop cannot decrease since it now has greater time to gather data from its predecessors.

This contradicts our assumption that W'_1, $W'_1 + 1$, ..., $W'_1 + n^*_1 - 1$ are not optimal transmission slots for C_1.

Assume that 1) and 2) are true for node C_m.

$j = m + 1$: If $\max(W^*_{m+1}, W'_m + n^*_m) = W^*_{m+1}$, then clearly W'_{m+1} is an optimal waiting time for node C_{m+1}. Suppose that $W^*_{m+1} < W'_m + n^*_m$. By 1) and 2) for node C_m, node C_{m+1} cannot start transmitting before $W'_m + n^*_m$. If node C_{m+1} waits until slot $W'_m + n^*_m (> W^*_{m+1})$, it can potentially aggregate more packets, and still make n^*_{m+1} transmissions. Therefore, the value of the objective function cannot decrease if C_{m+1} waits until $W'_m + n^*_m$. Hence, 1) follows for node C_{m+1}.

The proof of 2) is virtually identical to that for the case $j = 1$.

Thus, by induction, 1) and 2) are true $\forall j \in \{1, 2, ..., k\}$. ■

Theorem V.1 shows that in order to find a collision-free optimal schedule, it is enough to know the optimal waiting time of each node, and the optimal number of time slots to be allocated for transmission over each link. Specifically, it shows that if the optimal waiting time of a node i is W_i, and the optimal number of time slots it is allocated is n_i, then the optimal collision-free schedule of i is the set of time slots $\{W_i, W_i + 1, ..., W_i + n_i - 1\}$.

We now show that finding the optimal waiting times, and the optimal number of time slots to be allocated involves solving a single server Job Interval Selection Problem (JISP) which is known to be MAX SNP-Hard. We do this by rewriting problem π_D in a recursive manner.

Let $X[i, W]$ represent the maximum aggregated information received at node i if node i waits for a time W. For any leaf node l, for any $W \in \{0, 1, ..., D - 1\}$, we have $X[l, W] = w_l \lambda_l$. (Here, $\lambda_l = 1$, if l is a source, and zero, otherwise). Recall that w_l is the information provided by l. Consider any hop with parent node P having k children, C_1, C_2,..., C_k. Then, for any W, $X[P, W]$ can be calculated recursively as

$$X[P, W] = w_P \lambda_P + \max_{\{W_{C_i}, n_{C_i P}\}} \sum_{i=1}^{k} X[C_i, W_{C_i}] f_{C_i P}(n_{C_i P}), \tag{4}$$

where $\{W_{C_i}, n_{C_i P}\}$ satisfy the constraints of problem π_D. Further, $X[S, D]$ provides an optimal solution to π_D.

The next step is to show that finding $X[i, W]$ for any non-leaf node i (with k children) and for a given W is equivalent to solving a Maximum Weighted Independent Set (MWIS) problem. Recall that an independent set is a set of vertices in a graph, no two of which have an edge between them. A Maximum Weighted Independent Set is an independent set such that the sum of the weights of the vertices is maximum. For the following results, we assume that $W \geq \sum_{j=1}^{k} \beta_{C_j i}$. Note that by replacing $W - \sum_{j=1}^{k} \beta_{C_j i}$ by $\max(0, W - \sum_{j=1}^{k} \beta_{C_j i})$, the following results will hold for any $W \in \{0, 1, 2, ..., D - 1\}$.

Lemma V.2. *Let C_1, C_2, ..., C_k be the children of node i that is not a leaf node. If W^* is the optimal waiting time of node i, then one of the optimal set of time slots during which the children transmit is given by $\{W^* - \sum_{j=1}^{k} \beta_{C_j i}, W^* - \sum_{j=1}^{k} \beta_{C_j i} + 1, ..., W^* - 1\}$.*

Proof: From Theorem V.1, we know that transmitting in consecutive slots is optimal. We now prove this lemma by contradiction.

Suppose that the set of slots given above is not optimal. This means that at least one of the children makes a transmission before $W^* - \sum_{j=1}^{k} \beta_{C_j i}$. Since the maximum total number of transmissions for all the nodes in the hop is given by $\sum_{j=1}^{k} \beta_{C_j i}$, no child node makes a transmission in at least one of the slots in the set above. If the child node that had transmitted before the slot $W^* - \sum_{j=1}^{k} \beta_{C_j i}$ had waited until this free slot, it could have potentially gathered more packets, and still made a successful transmission, thus increasing $X[i, W]$. This contradicts the assumption that the above set of time slots is not optimal. ■

Theorem V.3. *$X[i, W]$ is obtained by finding a Maximum*

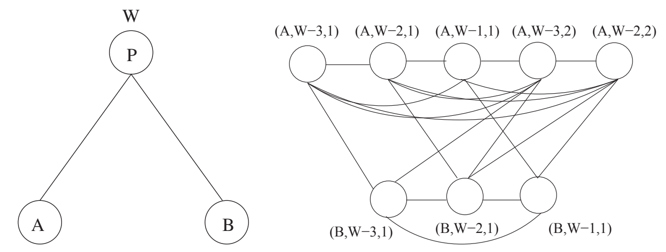
Weighted Independent Set (MWIS) in the interference graph, G', constructed as follows. For each C_j, $j \in \{1, 2, ..., k\}$, *for* each $n_{C_j i}$, $n_{C_j i} \in \{0, 1, 2, ..., \beta_{C_j i}\}$, *construct nodes labeled* $(C_j, W - \sum_{j=1}^{k} \beta_{C_j i}, n_{C_j i})$, $(C_j, W - \sum_{j=1}^{k} \beta_{C_j i} + 1, n_{C_j i})$, ..., $(C_j, W - n_{C_j i}, n_{C_j i})$, *respectively. The first term in the label represents the child, the second term represents the waiting time of the child, and the third term represents the number of transmissions made by the child. A node labeled* $(C_j, W_{C_j}, n_{C_j i})$ *in this graph is assigned a weight* $X[C_j, W_{C_j}](f_{C_j i}(n_{C_j i}))$. *For any two nodes* (C_y, W_y, n_y) *and* (C_z, W_z, n_z) *in* G'*, there exists an edge if and only if* (a) $\{W_y, W_y + 1, ..., W_y + n_y - 1\} \cap \{W_z, W_z + 1, ..., W_z + n_z - 1\} \neq \emptyset$, *or* (b) $C_y = C_z$.

Proof: From the construction of G', two nodes will have an edge either if they represent the same children, or the time slots during which the corresponding two children are scheduled are conflicting (since in the one-hop interference model, two children cannot simultaneously transmit to the same parent). Further from Lemma V.2, no child needs to transmit before $W - \sum_{j=1}^{k} \beta_{C_j i}$. This implies that an independent set in G' will have the following properties.

- Each child will be represented in the independent set at most once.
- The transmission schedules of no two children in the independent set conflict with each other.

$X[i, W]$ is thus obtained by finding a MWIS in G'. ■

Figure 2(a) illustrates a hop with two nodes, A and B, having parent P. Assume that $\beta_{AP} = 2$, and $\beta_{BP} = 1$. Then, for a waiting time W for P, the interference graph for these two children is shown in Figure 2(b). Note that a node having label A as the first term interferes with all other nodes having A as their first term. The same holds for B. This means that in an independent set only one of the nodes having the same first term can be chosen, i.e., a child can have only one schedule. Further, if the schedules of A and B conflict, then there are edges between these conflicting schedules in the interference graph. This can also be seen in Figure 2(b).



(a) Parent P with two children

(b) Interference graph

Fig. 2.

We now show that finding a MWIS in G' is NP-Hard. We briefly review the single server JISP [12], [13]. In this problem, k jobs need to be served by a single server. Each job has r instances, where each instance is associated with an explicit time interval during which it must be scheduled, and a certain profit for the instance. The server can serve only one instance of any job during each time slot. The goal is to find a schedule such that at most one instance of a job is present in the schedule, the instances in the schedule do not conflict in time, and the sum of the profits of the job instances is maximum.

Finding $X[i, W]$, i.e., an MWIS in G', is a JISP. This is because each job j in the JISP corresponds to the child C_j of node i. The set of instances of the job j corresponds to the set of nodes in G' whose first term is C_j. The schedule for each instance of the job is given by the second and third terms of the corresponding node in G' which represent the interval starting from the waiting time, and with a length equal to the number of time slots allocated to the child. The profit of each instance is the weight of the corresponding node in G'.

Thus, finding a MWIS in G' is identical to solving JISP, and hence *problem π_D is MAX SNP-Hard*. JISP is a well-studied integer programming problem [13], [12], and there exists $\frac{1}{2}$-approximation algorithms for special cases of JISP. However, JISP is a relatively small part of our problem. While finding $X[i, W]$ for each node i and for each W is a JISP, we ultimately need $X[S, D]$. If we use existing JISP approximation algorithms, they would result in a very poor performance (since if there are h hops in the tree, the approximation factor of the overall problem could be as poor as $\frac{1}{2^h}$). Therefore, we take these issues into account, and develop a new solution.

VI. Sub-optimal Formulation and Solution

The structural properties of π_D imply that in any hop the next node starts transmitting only after the previous node has finished transmitting. This is because the optimal schedule for each child in that hop is an interval, and the schedules of any two children cannot conflict, i.e., no two intervals can intersect. Therefore, in each hop, there is an order in which children are allocated time slots for transmission. The idea of the sub-optimal formulation is that we assume that in any hop, the order in which children transmit to their parent is known. We denote this problem π_D^{sub}. It turns out that once the order of transmission in each hop is known, the resulting problem can then be solved in polynomial time. Note that we still need to determine the waiting times and the number of time slots allocated to each node. For instance, consider two children C_1 and C_2. They have waiting times W_1 and W_2, respectively, and are allocated n_1 and n_2 time slots, respectively. Suppose we know that C_1's schedule is *before* C_2's schedule. Then, from Theorem V.1, we know that $W_1 + n_1 - 1 < W_2$. We now need to determine W_1, n_1, W_2, and n_2, with the additional constraint that the order in which children are scheduled is known.

We now provide some graph theoretic prelimiaries required to solve π_D^{sub}.

A. Preliminaries

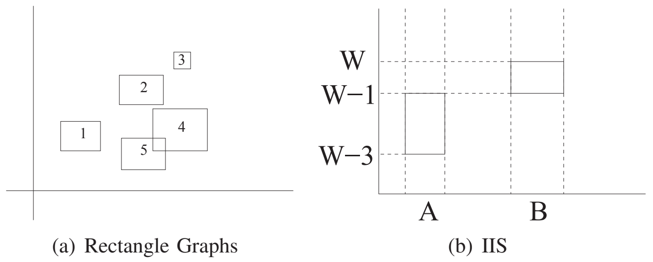
1) **Interval graph:** Let $\{I_1, I_2, ..., I_n\}$ be a set of intervals on the real line. Then, the interval graph $G(V, E)$ cor-

responding to this set of intervals is defined as follows.

- $V = \{I_1, I_2, ..., I_n\}$. Each vertex denotes an interval.
- For any $y, z \in \{1, 2, .., , n\}$, $(I_y, I_z) \in E$ if and only if the intervals intersect, i.e., $I_y \cap I_z \neq \emptyset$.

2) **Interval graph of interval number m:** The definition is identical to that of the interval graph except that each vertex can now be represented as a disjoint union of m intervals. An interval graph of interval number 2 is called a *double interval graph*.
3) **Rectangle graphs:** Rectangle graphs are a subclass of double interval graphs. A double interval graph can be transformed into a rectangle graph by simply labeling the vertices in the double interval graph as the set-product of the two intervals instead of the union of the two intervals. Thus, each vertex now represents a rectangle in $\mathbb{R}^2$. It is important to note that two rectangles that do not intersect need not form an independent set in the corresponding double interval graph. On the other hand, every independent set in the double interval graph is an independent set in the rectangle graph.
4) Maximum Weight Independent Set on interval graphs (order 1) can be found in polynomial time. However, MWIS on interval graphs of order m, $m > 1$, is still NP-Hard [14].
5) **Increasing Independent Set (IIS) on rectangle graphs:** An Increasing Independent Set (IIS) on a rectangle graph is an independent set that has the following property. Let $A = \{r_1, r_2, ..., r_m\}$ be an ordered set of rectangles ordered in the following fashion. For any $i, j \in \{1, 2, ..., m\}$ such that $i < j$,
   - The maximum x-coordinate of any point in $r_i \leq$ the minimum x-coordinate of any point in r_j.
   - The maximum y-coordinate of any point in $r_i \leq$ than the minimum y-coordinate of any point in r_j.

   Then, A is an IIS on the given rectangle graph. Note that the rectangles in A are ordered such that the next rectangle is *to the right and to the top* of the previous rectangle in the order. Further, an increasing independent set on a rectangle graph is an independent set on the corresponding double interval graph.



(a) Rectangle Graphs (b) IIS

Fig. 3. Rectangle Graphs and Independent Sets

We now provide an example to illustrate the definitions above. Consider Figure 3(a). Each rectangle in the figure represents a vertex in a rectangle graph. There will be an edge between two vertices only if the corresponding two rectangles intersect. For instance, there is an edge between rectangles 4 and 5. $\{1, 2, 3, 4\}$ and $\{1, 2, 3, 5\}$ are two maximal independent sets of rectangles. While $\{1, 4\}$ is an independent set in the rectangle graph, it does not form an independent set in the corresponding double interval graph because its intervals on the y-axis intersect. Also, $\{1, 2, 3\}$ is an example of an Increasing Independent Set (IIS) in the rectangle graph because rectangle 2 is to the right and to the top of rectangle 1, and rectangle 3 is to the right and to the top of rectangle 2.

Figure 3(b) shows an IIS for the example illustrated in Figure 2. Assuming that A transmits before B, an IIS is given by A transmitting from $W - 3$ to $W - 1$, and B transmitting from $W - 1$ to W. Clearly, the rectangle for B is to the top and to the right of the rectangle for A.

B. Solution

We show that when the order of schedules of children in any hop of the tree is known, finding $X[\cdot, \cdot]$ is equivalent to finding a Maximum Weighted Increasing Independent Set (MWIIS) on a rectangle graph.

Theorem VI.1. *Consider any hop with parent i having k children C_1, ..., C_k. WLOG, assume that the order in which the children are scheduled is $C_1 \rightarrow C_2 \rightarrow ... \rightarrow C_k$. Assign an interval $[a_i, b_i]$ to child C_i, $i \in \{1, 2, ..., k\}$, such that $W < a_1 < b_1 < a_2 < b_2 < ... < a_k < b_k$, and replace the first term in the label of each node in G' by this interval. Then, G' (defined in Theorem V.3) is a double interval graph, and $X[i, W]$ can be obtained by finding a Maximum Weighted IIS in the rectangle graph corresponding to G'.*

Proof: Assign an interval $[a_i, b_i]$ to child C_i, $i \in \{1, 2, ..., k\}$, such that $W < a_1 < b_1 < a_2 < b_2 < ... < a_k < b_k$. Replace C_i by $[a_i, b_i]$ in the first term of the label of each node in G'. Note that the second and the third terms in the label of each node of G' is an interval which specifies the time slots during which the child transmits. Each node in G' can now be represented as the union of two disjoint intervals, the first interval corresponding to the child, and the second interval corresponding to the time slots during which it transmits. Further, for any two nodes in G' that represent the same child, there exists an edge between the two nodes since the first interval in both the node labels have a non-empty intersection. Therefore, G' is a double interval graph.

G' can now be transformed into a rectangle graph as defined before. Let the x-axis represent the children, and the y-axis represent the schedules. Consider any two children, C_l and C_m. Let $l < m$, and hence C_l transmits before C_m. In the rectangle graph, a non-conflicting schedule for C_l and C_m can be achieved if and only if the rectangle corresponding to C_m's schedule is to the top and to the right of the rectangle corresponding to C_l's schedule. It cannot be to the left because $a_m > b_l$. It cannot be to the bottom because that would contradict the assumption that C_l transmits before C_m.

Thus, $X[i, W]$ can be obtained by finding a Maximum Weighted IIS in this rectangle graph corresponding to G'. ■

In [15], an algorithm has been proposed to determine a Maximum Weighted IIS in a rectangle graph for determining similarities in DNA sequences. For n rectangles in the rectangle graph, the complexity of this algorithm is $O(n \log n)$. G' has $O(k^2)$ vertices. Therefore, the complexity of finding $X[i, W]$ in Theorem VI.1 is $O(k^2 \log k)$.

So far, we have only provided an algorithm for calculating $X[i, W]$ for a particular node i, and a particular waiting time W. We ultimately need a procedure to find $X[S, D]$. Algorithm 1 (Table I) can be used for this purpose.

TABLE I
ALGORITHM 1

1	Start from the leaves. For a leaf node l, for each $W \in \{0, 1, ..., D-1\}$, $X[l, W] = \lambda_l w_l$.
2	For any non-leaf node i, for each $W \in \{0, 1, ..., D-1\}$, calculate $X[i, W]$ by finding an MWIIS in the rectangle graph corresponding to G' (Theorem VI.1).
3	Finally, calculate $X[S, D]$ at the sink. Use this to assign waiting times, and time slots to the sink's children.
4	Proceed from the sink's children down to the leaves. In any hop, once the waiting time of the parent P has been obtained (from its parent), the parent can assign waiting times and time slots for its children by looking up $X[P, \cdot]$.
5	Finally, assign waiting times and time slots to the leaves.

Theorem VI.2. *Algorithm 1 provides a collision-free schedule, and is optimal for the problem* π_D^{sub}.

Proof: This result follows from the previous results in the paper. ■

The computational complexity of Algorithm 1 for a tree with a maximum of h hops where each hop has k children on average can easily be calculated to be $O(hDk^2 \log k)$, where D is the deadline. One can thus see that the sub-optimal version has a very low computational complexity. The optimal solution for problem π_D requires an additional $O(k!)$ complexity for optimally ordering the children, since k children can be arranged only in $k!$ ways. Hence, the computational complexity of the optimal solution for π_D is given by $O(hDk!k^2 \log k)$.

To summarize, we make the following observations.

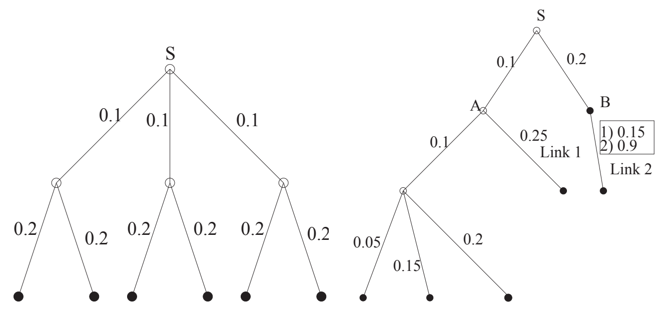
- Problem π_D is MAX SNP-Hard. However, in our problem, the exponential complexity ($k!$) is in the number of children (k) in a hop.
- If k is small (which is typically the case), then with $O(hDk!k^2 \log k)$ complexity we can solve problem π_D. Otherwise, we can use the sub-optimal version that has a complexity $O(hDk^2 \log k)$.

C. Discussion

In certain tree structures, the order in which children are scheduled does not affect the optimal solution of problem π_D.

1) **Single hop tree networks:** Since all nodes apart from the sink are leaf nodes, only the sink performs in-network computation. Therefore, the order in which leaf nodes are scheduled does not matter. Hence, the sub-optimal solution is optimal in this case.
2) **Symmetric trees:** A symmetric tree is defined as one in which nodes that are equal number of hops away from the sink satisfy the following:
    a) They have equal number of children.
    b) Either all of them are source nodes or none of them are source nodes, and they provide the same amount of information.
    c) Each incoming link has the same link reliability function.

An example of a symmetric tree is given in Figure 4(a). It is easy to see that the order of transmission of children in any hop of a symmetric tree does not affect the optimal solution to problem Π_D.



(a) Symmetric Tree (b) General Tree

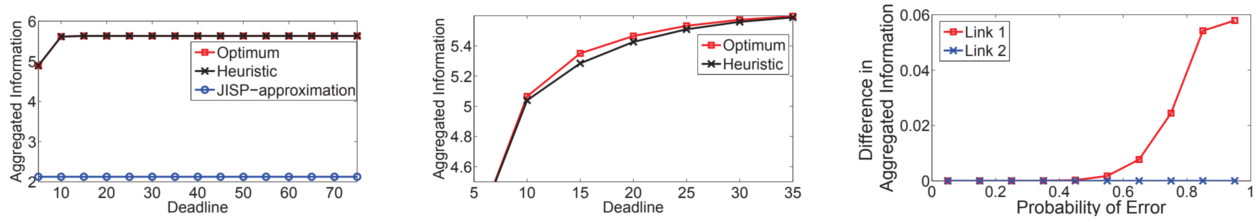
Fig. 4.

Since the only difference between the sub-optimal and the optimal problem is in determining the order in which nodes transmit in a hop, one can use intelligent heuristics to come up with a particular policy for scheduling. However, in general, choosing the optimal order of transmission in any hop depends on a number of factors such as the number of source nodes in the sub-tree, the entire structure of the sub-tree, and the link errors in the sub-tree.

VII. Numerical Results

In this section, we numerically investigate the performance of the sub-optimal solution with the optimal solution for general trees. For the purpose of ordering nodes in the sub-optimal solution, we order nodes (having the same parent) to transmit such that a node with a greater number of source nodes in its sub-tree transmits later than a node with a lesser number of source nodes in its sub-tree. We call this heuristic H. We use retransmissions as the error-recovery scheme for these simulations. The link reliability of a link (i, j) is given by $f_{ij}(n_{ij}) = (1 - p_{ij}^{n_{ij}})$, where p_{ij} is the probability of error over link (i, j).

We consider the tree in Figure 4(b) with link error probabilities as shown and source nodes represented by filled circles. We consider two trees with the same structure but with one of the links (Link 2 in Figure 4(b)) having a different probability of error (shown in a box in Figure 4(b)). The deadline is varied from 5 to 75. For each link (i, j), we select β_{ij} such that the probability that a packet gets lost in all the β_{ij} transmissions is less than 0.01.



(a) Link 2 with 0.15 error probability (b) Link 2 with 0.9 error probability (c) Difference in aggregated information

Fig. 5. Numerical evaluations

We first set the probability of error of Link 2 to be 0.15. From Figure 5(a), we can see that heuristic H is optimal for this tree with the given link error probabilities. Note that the JISP approximation algorithm in [13] performs very poorly relative to our algorithms. This is because JISP is only a part of our problem, and these algorithms cannot be directly applied to our problem since we have a multi-hop network.

We now change the probability of error of Link 2 to 0.9. Now from Figure 5(b), we observe that heuristic H is actually not optimal for certain deadlines. This is because for these deadlines the optimal solution is for B to transmit before A. However, when the deadline is small or large, the sub-optimal solution is optimal. This can be reasoned as follows. When the deadline is small (less than 10 time slots), the amount of aggregated information that is obtained from node B is not much since B accounts for only two nodes. Since A accounts for four nodes, it is optimal for A to wait longer than B to gather packets from its predecessors, and transmit after B finishes transmitting. However, when the deadline is slightly larger, A would have gathered packets by a certain time slot, which occurs much before the deadline. However, since the link from B's child has a high probability of error, B might not have gathered its child's packet. Therefore, if A was scheduled before B, B could have waited longer to give more time to its child to transmit its packet. Therefore, the optimal order of transmission here is to transmit B after A. Finally, when the deadline is very large, the order of scheduling of A and B no longer matters because each have sufficient time to gather packets from their predecessors. Note that we do not show the JISP algorithm in Figure 5(b) since the JISP provides an aggregated information < 2, and the difference between the sub-optimal and the optimal solution is very small that the difference cannot be observed if the JISP algorithm is included.

Furthermore, we observe that even if some links are bad, the heuristic can still be close to optimal. For this experiment, we vary the error probability of Link 1 from 0.05 to 0.95 while keeping that of Link 2 fixed, and vice versa. We fix the deadline to be 15 time slots. From Figure 5(c), it is clear that irrespective of the error probability of Link 1, the sub-optimal solution is still optimal. However, for Link 2, as the error probability increases above 0.35, the sub-optimal solution begins to deviate from the optimal solution. Therefore, high error probabilities do not necessarily change the optimal order of transmission.

VIII. Conclusion

In this paper, we have studied the problem of maximizing aggregated information in tree networks with unreliable links in the presence of deadline constraints. We formulated an integer programming problem that explicitly accounted for interference, link errors, and deadlines. We showed that the integer programming problem was MAX SNP-Hard, and looked at a sub-optimal version. We provided a low complexity, distributed optimal solution to the sub-optimal version, and analyzed tree structures for which the sub-optimal solution was actually optimal. Further, we studied the performance of our algorithm for arbitrary tree structures through simulations, and understood when it performs optimally and when it does not. Future work involves extensions to general interference constraints, and also to explicitly consider energy constraints.

References

[1] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient Communication Protocols for Wireless Microsensor Networks," in *International Conference on System Sciences*, 2000.
[2] Y. Wu, S. Fahmy, and N. B. Shroff, "On the construction of a maximum-lifetime data gathering tree in sensor networks: Np-completeness and approximation algorithm," in *IEEE INFOCOM*, 2008.
[3] A. Goel and D. Estrin, "Simultaneous optimization for concave costs: Single sink aggregation or single source buy-at-bulk," in *SODA*, 2003.
[4] H. L. V. Trees, *Detection, Estimation, and Modulation Theory: Part I.* John Wiley & Sons, 1968.
[5] L. Gargano and A. A. Rescigno, "Optimally fast data gathering in sensor networks," *Lecture Notes in Computer Science*, vol. 4162, 2006.
[6] C. Florens, M. Franceschetti, and R. McEliece, "Lower bounds on data collection time in sensory networks," *IEEE JSAC*, vol. 22, no. 6, 2004.
[7] B. Krishnamachari, D. Estrin, and S. B. Wicker, "The impact of data aggregation in wireless sensor networks," in *ICDCSW '02*, 2002.
[8] A. Boulis, S. Ganeriwal, and M. B. Srivastava, "Aggregation in sensor networks: An energy-accuracy trade-off," in *1st IEEE Int'l. Wksp. on Sensor Network Protocols and Applications*, 2003.
[9] Y. Yu, B. Krishnamachari, and V. K. Prasanna, "Energy-latency tradeoffs for data gathering in wireless sensor networks," in *IEEE INFOCOM*, 2004.
[10] Z. Ye, A. Abouzeid, and J. Ai, "Optimal policies for distributed data aggregation in wireless sensor networks," in *IEEE INFOCOM*, 2007.
[11] S. Hariharan and N. B. Shroff, "Maximizing aggregated revenue in sensor networks under deadline constraints," *IEEE CDC*, 2009.
[12] J. Chuzhoy, R. Ostrovsky, and Y. Rabani, "Approximation algorithms for the job interval selection problem and other related problems," *Mathematics of Operations Research*, vol. 31, 2006.
[13] F. C. R. Spieksma, "On the approximability of an interval scheduling problem," *Journal of Scheduling*, vol. 2, no. 5, 1999.
[14] R. Bar-Yehuda and M. M. Halldórsson, "Scheduling split intervals," *SIAM J. Comput.*, 2006.
[15] D. Joseph, J. Meidanis, and P. Tiwari, "Determining dna sequence similarity using maximum independent set algorithms for interval graphs," *Lecture Notes in Computer Science*, 1992.